Postprint version of paper published in J. Process Control 2015, 26:90–101
The content is identical to the published paper, but without the final typesetting by the publisher.

Improving classification-based diagnosis of batch processes through data selection and appropriate pretreatment

Geert Gins[a], Pieter Van den Kerkhof[a], Jef Vanlaer[a], Jan F.M. Van Impe[a,*]

[a] *Chemical and Biochemical Process Technology and Control (BioTeC), Department of Chemical Engineering, KU Leuven, W. de Croylaan 46, B-3001 Leuven, Belgium*

Abstract

This work considers the application of classification algorithms for data-driven fault diagnosis of batch processes. A novel data selection methodology is proposed which enables online classification of detected disturbances without requiring the estimation of unknown (future) process behavior, as is the case in previously reported approaches.

The proposed method is benchmarked in two case studies using the `Pensim` process model of Birol et al. (2002) implemented in `RAYMOND`. Both a simple k Nearest Neighbors (k-NN) and complex Least Squares Support Vector Machine (LS-SVM) are employed for classification to demonstrate the generic nature of the proposed approach. In addition, the influence of different data pretreatment methods on the classification performance is discussed, together with a motivation for selecting the correct pretreatment steps. Finally, the influence of the number of available training batches is studied.

The results demonstrate that a good classification performance can be achieved with the proposed data selection method even with a low number of faulty training batches by exploiting knowledge on the nature of the to-be-diagnosed faults in the data pretreatment. This provides a proof of concept for classification-based batch diagnosis and demonstrates the importance of

*Corresponding author

Email addresses: `geert.gins@cit.kuleuven.be` (Geert Gins), `pieter.vandenkerkhof@cit.kuleuven.be` (Pieter Van den Kerkhof), `jef.vanlaer@cit.kuleuven.be` (Jef Vanlaer), `jan.vanimpe@cit.kuleuven.be` (Jan F.M. Van Impe)

Journal homepage: http://www.journals.elsevier.com/journal-of-process-control/
Original file available at: http://www.sciencedirect.com/science/article/pii/S0959152415000177

incorporating process insight in the construction of data-driven process monitoring and diagnosis tools.

Keywords: Batch processes, Fault detection/isolation, Process control, Mathematical modeling, Fault classification

1. Introduction

Batch processes play an important role in the chemical and biochemical industries for the production of goods with a high added value (e.g., pharmaceuticals, food products, polymers, semiconductors) owing to their lower capital cost and higher flexibility to produce multiple products or grades. As any industrial process, batch processes can be prone to a number of disturbances such as impurities in the raw materials, fouling of heat exchangers, sensor failures, plugged pipes, etc. Since online product quality measurements are rarely available, close monitoring of batch processes and fast Fault Detection & Isolation (FDI) are absolute requirements to avoid unnecessary variations and ensure the final products are within specifications. The dynamic nature of batch processes further complicates FDI.

Today's process plants dispose of large historical databases containing measurements from hundreds of online sensors. Statistical Process Monitoring (SPM; sometimes referred to as Statistical Process Control) aims to exploit these historical data for FDI.

Most recent research in the field of FDI/SPM for batch processes has been devoted to fault detection and identification using *latent variable* approaches based on Principal Component Analysis (PCA) or Independent Component Analysis (ICA). While progress has been made in improving fault *detection* by including process dynamics (e.g., dynamic PCA [1], auto-regressive PCA [2], dynamic ICA [3]) or nonlinear extensions of PCA (e.g., kernel PCA [4], kernel ICA [5]), correct *isolation* (diagnosis) of the detected disturbance remains a difficult issue [6].

Contribution plots [7, 8], which chart the contribution of each variable to an out-of-control signal, are by far the most popular tool to identify the root cause of an alarm signal in batch monitoring. The generation of contribution plots requires no prior knowledge about disturbances. However, process insight is necessary for interpreting the contribution pattern and finding the root cause. Moreover, Westerhuis et al. [9] and Van den Kerkhof et al. [10]

illustrated that contribution plots can yield misleading results due to the so-called *fault smearing effect*.

If historical data of the different *known* faults types are available, diagnosis reduces to classification. A classifier is trained on faulty data and, upon detection, it assigns the detected fault to the class it most resembles. The time-consuming interpretation of contribution plots to find the root cause is eliminated, which significantly reduces the response time between fault detection and subsequent corrective action(s).

For continuous processes in steady state, various classification-based diagnosis methods using, e.g., discriminant partial least squares [11], Fisher discriminant analysis [12], support vector machines [13, 14] and neural networks [15] were reported in literature. For batch processes however, reports on the successful application of classification techniques are scarce. Since process plants are monitored and controlled to achieve satisfactory product quality and prevent process faults, the number of available faulty batches is limited. This is an important consideration for the design of a data-driven fault diagnosis scheme and forms an important bottleneck in the development of data-driven FDI.

Cho and Kim [16] proposed a Fisher Discriminant Analysis (FDA)-based classifier, but required a number of past faulty batches greater than the dimensionality of the fault data. In their case study, they simulated as many as 3500 faulty training batches for a process where 11 variables are monitored over 241 time points. Cho and Kim [17] generated pseudo-batches to deal with the data insufficiency. Cho [18] handled the data insufficiency more efficiently by extending linear FDA to nonlinear problems by employing *kernel* FDA. Li and Cui [19] further reduced the need for pseudo-batch generation. Recently, Support Vector Machines (SVMs) were utilized as a learning algorithm for fault classification of batch processes [20]. SVMs are based on statistical learning theory developed by Vapnik [21] and have shown to exhibit a large generalization performance, especially when the number of training samples is small [22]. Hence, SVMs are well suited for classification-based FDI.

The aforementioned methods [16–20] are essentially offline diagnosis methods since the classifier is trained on entire faulty batches, not faulty episodes. Hence, before fault classification, the new (faulty) batch needs to be completed or its future variable trajectories estimated. This is a major drawback. Moreover, the similarity between faulty batches with exactly the same fault at the same time instance decreases over time due to nonlinear process dy-

namics, different corrective actions and additional faults.

Therefore, this paper proposes a new fault diagnosis methodology which focuses on the onset of the fault rather than the entire batch history by sending only a small data window at the time of detection to the classifier. This eliminates the need for estimating future process behavior. The proposed method is validated on data generated from the `Pensim` model developed by Birol et al. [23], a widely-used benchmark for data-driven FDI. To demonstrate the general applicability of the proposed method, both a simple k Nearest Neighbors (k-NN) [24] and complex Least Squares Support Vector Machine (LS-SVM) [25] classifier are used. Several data pretreatment steps are proposed to improve classifier performance, together with guidelines for selecting the appropriate steps. Additionally, the influence of the number of available training batches on the method's performance is studied.

The remainder of the paper is organized as follows. First, the proposed fault diagnosis methodology is outlined in Section 2. Section 3 describes the case study on which the fault diagnosis method is validated. Sections 4 and 5 briefly summarize the basics of standard PCA-based batch monitoring, which is used as the fault detection method in this work, and fault identification using k-NN or LS-SVM classifiers, respectively. They are followed by a discussion of the validation procedure in Section 6 and the obtained results in Section 7. Finally, conclusions and future research directions are provided in Section 8.

2. Proposed fault diagnosis method

Section 2.1 presents an overview of the proposed classification-based diagnosis method while Section 2.2 delves deeper into the critical issue of data pretreatment.

2.1. Overview

The proposed diagnosis methodology entails two phases: (i) an offline model building phase and (ii) an online diagnosis phase. A general scheme of the method is presented in Fig. 1.

2.1.1. Offline model building

The first step of the offline model building phase consists of scanning past batches for faulty episodes by means of an appropriate fault detection model (e.g., an NOC PCA model and corresponding fault detection statistics). By

Postprint version of paper published in J. Process Control 2015, 26:90–101
The content is identical to the published paper, but without the final typesetting by the publisher.

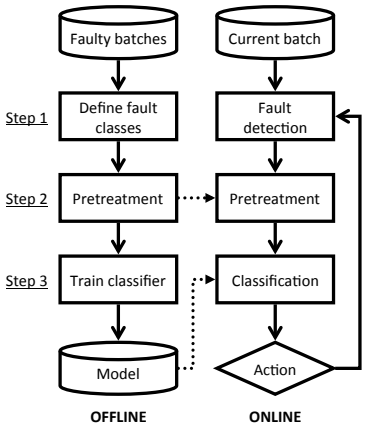

Figure 1: Scheme of the proposed batch diagnosis method.

combining process knowledge and operator experience, the root cause of each detected fault is investigated. Based on this study, fault classes are defined and the detected faults are assigned to one of these fault classes. The definition of fault classes is an important step. A large number of fault classes reduces classification performance due to a decreasing amount of training batches per class and an increasing similarity between different classes. On the other hand, coarsely defined classes are unhelpful for straightforward corrective actions. Therefore, in practice the number of classes is a trade-off between classifier performance and practical use of the diagnosis results.

During the second step, the raw data are converted into a form amenable for learning a classification model (e.g., a k-NN or LS-SVM classifier). The goal of data pretreatment is to obtain a uniform characteristic fault pattern for each fault class. Data pretreatment is crucial for the classifier's diagnosis performance.

After collecting the data and converting these into a form suited for classification, the classifier is trained.

2.1.2. Online fault diagnosis

During the online or application phase, the current batch is monitored using the existing fault detection system. If a fault is detected, the data of the current batch undergo the same pretreatment steps as the faulty reference data. The pretreated data are subsequently passed to the trained classifier which assigns a fault class to the current disturbance. Based on this infor-

Journal homepage: http://www.journals.elsevier.com/journal-of-process-control/
Original file available at: http://www.sciencedirect.com/science/article/pii/S0959152415000177

mation, operators take corrective actions or abort the process if the batch cannot be salvaged. Monitoring the fault detection statistics reveals if the operator actions bring the process back within normal operating conditions. If the diagnosis result was satisfactory, the data can be added to the reference set of faults and the classifier retrained.

2.2. Data selection and pretreatment

It is a well-known fact that a classifier's performance depends heavily on the training data. Raw process data are seldom in an appropriate form for training a classifier. Transforming raw data into a suitable form is typically a time consuming step in any data-based diagnosis project and involves trial and error. Moreover, while classification algorithms are most often general-purpose, data selection and pretreatment are highly application specific. This section proposes several selection and pretreatment steps for batch process data. Their impact on diagnosis performance is analyzed in Section 7.

Dimensionality reduction is a first possible pretreatment step. Monroy et al. [20] reduced the data of each batch to a score vector using PCA or ICA and use the resulting scores as input for the classifier. To train the PCA or ICA model, Monroy et al. [20] generated artificial data sets containing an equal number of normal and faulty batches. In practice however, the number of faulty batches is much smaller than the number of normal batches, making the composition of a balanced and sufficiently large training set for the PCA or ICA model a challenging issue or even infeasible. In order to propose a widely applicable diagnosis method, reduction of the classifier's training data is avoided in this work and all pretreatment steps occur in the original measurement space.

The amount of data per faulty batch to include in the training set, i.e., data selection, is a second important issue to consider. Previously reported methods for classification-based diagnosis typically include each entire faulty batch. In Monroy et al. [20], the scores are computed from the completed faulty batch. Cho and Kim [16] also train their classifier on the full data of each faulty batch. As a consequence, the future data of the current batch need to be predicted to enable online diagnosis. Cho and Kim [16] solved this problem by selecting the past batch trajectory most similar to the current batch to estimate future measurements.

Including each entire faulty batch is seldom a good choice in view of obtaining a uniform characteristic fault pattern for each fault. Consider two batches with a fault of exactly the same type and magnitude but different

Postprint version of paper published in J. Process Control 2015, 26:90–101
The content is identical to the published paper, but without the final typesetting by the publisher.

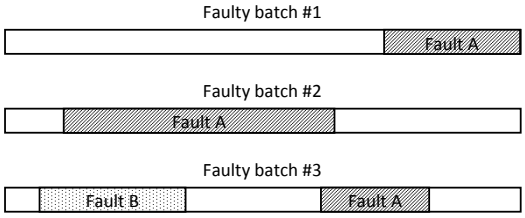

Figure 2: The similarity between faulty batches with the same fault decreases due to different onset times (batches 1 and 2) and additional faults (batch 3). Therefore, data selection by focusing on each faulty episode is a better approach to obtain a uniform characteristic fault pattern than including each entire faulty batch in the analysis.

onset times, such as batches 1 and 2 in Fig. 2. The same fault occurs near the end of the first batch and at the start of the second batch. In the approach of Monroy et al. [20], the data of each entire batch are transformed to a score vector. Ideally, since the two batches exhibit the same fault, both score vectors are close to identical. However, since the fault in the first batch occurred near the end, the largest part of this batch is NOC data and hence the corresponding score vector is likely more similar to NOC scores than it is similar to the second faulty batch's scores. The similarity between the two faulty batches can decrease even further due to different corrective actions, additional faults at other time points (e.g., batch 3 in Fig. 2), etc.

The similarity between faulty batches of the same class is highest at the onset of the fault. Therefore, the authors propose to focus data selection on the faulty episode by only retaining in the training set a small data window starting from the moment of fault detection. Different choices of the data window are possible. A first option is to include only the $1 \times J$ measurement vector at the time of detection, where J is the number of sensors considered in the fault detection system. The advantage of this approach is that as soon as a new fault is detected, the current measurement vector can be passed to the classifier to obtain the diagnosis result. On the other hand, using only the J measurements at detection sometimes provides insufficient discrimination between similar faults. A second option is to add measurements at $N-1$ later time points in a $1 \times JN$ vector. In this case, diagnosis of a new batch is only possible after these measurements are available, but classification performance might increase. A missing data estimator can be added to overcome the latter problem. The effect of the time window length

Journal homepage: http://www.journals.elsevier.com/journal-of-process-control/
Original file available at: http://www.sciencedirect.com/science/article/pii/S0959152415000177

on classification performance is illustrated in Section 7.

The last step concerns data transformations which can range from a simple variable scaling to more complex functions. The choice of a suitable transformation depends heavily on process knowledge. For batch process data, normalization around the mean trajectory is a common transformation before constructing fault detection models. Normalizing around the mean trajectory is also a valuable pretreatment step for fault classification as it substantially reduces the difference between faults of the same class occurring at differing time points. Section 7 demonstrates the impact of normalization and other scaling methods. In addition, it provides a motivation for choosing specific normalization or scaling methods.

3. Case study

As a case study, data of a simulated industrial-scale biochemical process for penicillin fermentation are generated, using an extended version of the `Pensim` simulator developed by Birol et al. [23]. This process is a widely used benchmark process for evaluating FDI methodologies.

The production process involves two phases: a batch phase and a fed-batch phase. Initially, the bioreactor is operated in batch mode. When the substrate concentration has decreased to 0.3 g/L (after about 43 h), the fed-batch phase is started, and additional substrate is continuously fed into the reactor. After adding 25 L of substrate (after approximately 460 h), the fermentation is stopped. During the fermentation, 11 online sensors record various flows, temperatures, and pH; the variables are listed in the first column in Table 1. Time is added as an extra (12th) variable. The measured signals are aligned and resampled to a length of 101 samples for the batch phase and 501 samples for the fed-batch phase, using the indicator variables proposed by [23].

Two case studies are conducted in this paper. The first (base) study is a strict implementation of `Pensim`, where sensor noise is only present on the DO and CO_2 concentration [23]. To obtain a more involved/advanced second study, white Gaussian noise with zero mean is added to all online sensors. The corresponding absolute or relative (to the sensor reading) standard deviations are listed in the second column in Table 1. Noise on the pH and reactor temperature directly influences the control actions on the acid, base, and coolant flow rates. To retain good control performance, the PID controllers were retuned. In addition, biological variability is added to

Postprint version of paper published in J. Process Control 2015, 26:90–101

Table 1: Overview of online measurements of the `Pensim` model. The second column lists the standard deviation of the measurement noise as used for the advanced case study.

Sensor measurement	σ_{noise}
DO concentration [mmol/L]	0.002
CO_2 concentration [mmol/L]	0.060
pH [-]	0.010
Reactor temperature [K]	0.050
Feed temperature [K]	0.050
Culture volume [L]	0.001
Agitator power [W]	0.5%
Substrate feed rate [L/h]	0.5%
Coolant flow rate [L/h]	0.5%
Base flow rate [L/h]	0.5%
Acid flow rate [L/h]	0.5%
Time [h]	—

the simulation. Hereto, a PRBS signal with magnitude 0.005 is averaged over 1000 time samples and added to the constant (0.092 h^{-1}) profile of the maximal specific biomass growth rate.

In each case study, 200 batches under normal operating conditions (NOC) with varying initial conditions are simulated as a reference set for the fault detection model.

For training of the fault classifiers, 1000 batches are generated for each of six faults listed in Table 2 for a total of 6000 faulty batches. The feed concentration and coolant temperature steps are present from the start of the batch. The agitator power and aeration rate drops are two sudden faults. Finally, the feed rate and DO sensor drifts are incipient faults that gradually build up over time. The starting time and magnitude of each fault are randomly chosen from a uniform distribution with the bounds stated in Table 2. The lower bounds are chosen as to achieve satisfactory fault detection performance. For faults where the magnitude can be positive or negative (e.g., drifts on the DO sensor) an equal number of batches of each sign is simulated.

Finally, an independent set of 600 faulty batches is available for validation of the classifiers (100 batches per fault class).

Journal homepage: http://www.journals.elsevier.com/journal-of-process-control/
Original file available at: http://www.sciencedirect.com/science/article/pii/S0959152415000177

Postprint version of paper published in J. Process Control 2015, 26:90–101
The content is identical to the published paper, but without the final typesetting by the publisher.

Table 2: Simulated fault classes and their magnitude and starting time ranges.

Fault type	Magnitude	Onset time [h]
Feed concentration step	$\pm[1\%; 10\%]$	0
Coolant temperature step	$\pm[1\%; 10\%]$	0
Agitator power drop	$[-5\%; -30\%]$	$20 - 380$
Feed rate drift	$\pm[0.15\%/h; 0.35\%/h]$	$70 - 380$
Aeration rate drop	$[-70\%; -90\%]$	$20 - 380$
DO sensor drift	$\pm[0.50\%/h; 0.75\%/h]$	$20 - 380$

4. Fault detection

In order to identify periods of faulty operation, either for building the classifier's training set or during online diagnosis, a fault detection method needs to be in place. Standard PCA-based fault detection is employed in this work because it is the standard benchmark for data-driven FDI [6]. It and its many variants are widely studied (e.g., [1, 2, 8, 26–34]). Detailed overviews of research and applications can be found in, e.g., Qin [6] or Ge et al. [35].

PCA-based fault detection is a combination of a PCA model of the collected batch data that characterizes the normal process operation (Section 4.1) and one or more fault detection statistics that detect deviations from normal operation (Section 4.2). Only a brief summary is presented here.

While PCA-based FDI is employed here, the authors would like to stress that the proposed diagnosis method is independent of the chosen fault identification technique.

4.1. Characterizing normal operating conditions

Industrial data, such as batch process data, are typically heavily correlated because all measured variables are connected through underlying physical laws, mass balances, redundancy of sensors, etc. PCA reduces the number of measured variables to a smaller number of uncorrelated latent variables or *scores* by exploiting these correlations [36, 37]. It is used to characterize the NOC reference data set, containing 200 batches in which 12 variables are measured at 602 different time points, resulting in a $200 \times 12 \times 602$ batch data array. Because the benchmark process consists of two phases with clearly different dynamics, a separate PCA model is constructed for each phase. The

Journal homepage: http://www.journals.elsevier.com/journal-of-process-control/
Original file available at: http://www.sciencedirect.com/science/article/pii/S0959152415000177

first phase consists of the first 101 time points; the final 501 time points make up the second phase.

The data of each of the 12 sensors are first normalized around their mean trajectory to zero mean and unit variance. This normalization removes most nonlinearities from the data and reduces the problem to analyzing the approximately linear process dynamics around the average trajectory [26, 38]. Hence, the use of a linear PCA model to characterize these normal operation data is justified.

After normalization, the data are unfolded using the variable-wise unfolding method proposed by Wold et al. [39]: the 101×12 and 501×12 measurements of the each batch phase p are placed under each other to obtain an unfolded data matrix $\mathbf{X}_p$ of size $20{,}200 \times 12$ for the first batch phase and $100{,}200 \times 12$ for the second batch phase. A separate PCA model for each phase summarizes each row of $\mathbf{X}_p$ using R scores ($R \leq 12$). The columns of the score matrix $\mathbf{T}_p$ (20,200 or 100,200 $\times$ R) are linear combinations of the original variables as defined by phase p's loading matrix $\mathbf{P}_p$ ($12 \times R$), of which each column corresponds to one of the R retained principal components of the data covariance matrix $\mathbf{X}_p^\top \mathbf{X}_p$. The scores can be seen as new uncorrelated variables [37]. The approximation residuals are contained in $\mathbf{E}_\mathbf{X}$ (20,200 or 100,200 $\times$ 12).

$$\mathbf{X}_p = \mathbf{T}_p \mathbf{P}_p^\intercal (+\mathbf{E}_{\mathbf{X},p}) \tag{1}$$

Jollife [36] provides an overview of criteria for selecting the number of principal components R. In this paper, the number of principal components is determined with an adjusted Wold criterion with a threshold of 0.90 on the fraction of explained variance [40]. Three principal components are selected for the batch phase and four for the fed-batch phase, explaining 60% and 64% of the total variance, respectively.

4.2. Fault detection statistics

Abnormal behavior is detected by comparing measured process data of a new, running batch against NOC data. A short summary of the detailed description given in Nomikos and MacGregor [27] is presented here.

At each time point t, the current measurement vector $\mathbf{x}_t$ (1×12) is projected on the current phase's loading matrix $\mathbf{P}_p$ to obtain the current score vector $\mathbf{t}_t$ ($1 \times R$) and residual vector $\mathbf{e}_t$ (1×12). These vectors are compared to the NOC data via fault detection statistics.

Postprint version of paper published in J. Process Control 2015, 26:90–101
The content is identical to the published paper, but without the final typesetting by the publisher.

Hotelling's T^2 statistic monitors the scores and checks if a new observation projects onto the model plane defined by the loading matrix $\mathbf{P}_p$ within the limits determined by the NOC data. The Squared Prediction Error (SPE) statistic monitors the residuals to detect the occurrence of any abnormal events that cause new observations to move away from the model plane. Upper control limits are established for both statistics based on the reference data set [26]. To mitigate the occurrence of false alarms, an alarm signal is raised only when a statistic is above its control limit at three or more consecutive time points.

Because variable-wise unfolding is employed in this work, only measurements from a single time point are required to compose a complete observation row of $\mathbf{X}$. Hence, running batches can be analyzed online without requiring the estimation of future (unknown) measurements.

5. Fault identification

Once faulty operation is detected, a classification model is used to identify the root cause of the upset. In the presented case studies, two types of classifiers are employed to demonstrate the proposed fault identification method: k-NN and LS-SVM.

The k-NN classifier (Section 5.1) is selected as a simple yet powerful classifier [41–43]. An LS-SVM (Section 5.2) is used as an example of a very powerful, nonlinear classification method because of its very good generalization properties given the limited availability of faulty batches for classifier training [22].

While k-NN and LS-SVM are employed in this paper, the purpose of this paper is not to identify the best classifier but rather to demonstrate the potential of the proposed fault identification methodology. For other applications, other classifiers might prove more optimal.

5.1. k Nearest Neighbors

The k-NN [24] method is one of simplest classification algorithms. Despite its simplicity, k-NN has proven to be a powerful classification tool [43, 44].

The training phase of a k-NN is trivial: it solely consists of storing each training sample and its corresponding label. The unknown sample $\mathbf{x}_t$ is assigned to the class that is most common among the k closest (most similar) data points according to a chosen distance measure. If the votes of the k neighbors tie, the point is considered unclassifiable.

Journal homepage: http://www.journals.elsevier.com/journal-of-process-control/
Original file available at: http://www.sciencedirect.com/science/article/pii/S0959152415000177

The appropriateness of the k-NN algorithm decreases for high-dimensional data as the distance to the nearest neighbor approaches the distance to the farthest neighbor [45]. This effect already occurs for data dimensionality as low as $10-15$. However, the good results obtained with PCA- or ICA-based FDI demonstrate that the *effective* dimensionality of industrial data is often small. (For example, Bezergianni and Kalogianni [46] describe a typical example where a month's worth of data on 38 process variables in a hydro treating process is governed by only 4 latent variables.) Hence, a k-NN classifier is a valid classification approach for fault identification.

In this paper, a basic version of the k-NN is implemented, which employs the Euclidean distance metric. Both $k = 1$ neighbor and $k = 3$ neighbors are employed.

5.2. Least Squares Support Vector Machines

The concept of LS-SVMs and their extension to multi-class problems is briefly discussed here. The interested reader is referred to the book of Suykens et al. [25] for a detailed treatment of LS-SVMs.

5.2.1. LS-SVMs

Consider a classifier training data set consisting of N samples of M-dimensional input data $\mathbf{x}_i$ $(i = 1 \dots N)$ belonging to two classes labeled with a scalar $y_i \in \{-1, +1\}$ for the positive and the negative classes, respectively. In their simplest form, LS-SVMs train a linear decision function or hyperplane from the input data and classify a new data point $\mathbf{x}_t$.

$$y\left(\mathbf{x}_t\right) = \operatorname{sign}\left(\mathbf{w}^\intercal \mathbf{x}_t + b\right) \tag{2}$$

$\mathbf{w}$ is an M-dimensional parameter vector and b a scalar bias term. LS-SVMs seek the hyperplane that maximizes the margin between the two classes. This maximum margin principle leads to a higher generalization performance, i.e., an increased correct classification rate for unseen data points. For non-separable cases, a regularization parameter reflects the trade-off between margin maximization and minimization of the classification error during identification of the parameters $\mathbf{w}$ and b.

LS-SVMs are extended to nonlinear classifiers by projecting the data to a higher-dimensional space using a nonlinear transformation function $\varphi(\cdot)$. Exploiting Mercer's theorem, Eq. (2) is rewritten using the kernel function $K(\mathbf{x}_n, \mathbf{x})$ that implicitly defines $\varphi(\cdot)$.

Postprint version of paper published in J. Process Control 2015, 26:90–101
The content is identical to the published paper, but without the final typesetting by the publisher.

$$y\left(\mathbf{x}_t\right) = \text{sign}\left(\sum_{n=1}^{N} \alpha_n y_n K(\mathbf{x}_n, \mathbf{x}_t) + b\right) \tag{3}$$

N is the number of training data points and $K(\mathbf{x}_n, \mathbf{x})$ is the kernel function of the nth vector of the training set and a new vector $\mathbf{x}_t$ [22].

LS-SVMs are a modification of the original SVMs. The training of an LS-SVM converts from a quadratic programming problem to solving a set of linear equations for finding the parameters α_n and b.

Substituting a nonlinear kernel function yields a nonlinear LS-SVM classifier. In this work, the simple linear kernel and the popular Radial Basis Function (RBF) kernels are employed.

$$K(\mathbf{x}_n, \mathbf{x}_t) = \mathbf{x}_n^{\mathsf{T}} \mathbf{x}_t \qquad \text{linear kernel}$$

$$K(\mathbf{x}_n, \mathbf{x}_t) = \exp\left(-\frac{\left\|\mathbf{x}_n - \mathbf{x}_t\right\|^2}{2\sigma^2}\right) \qquad \text{RBF kernel}$$

where σ is a parameter called the kernel width. Other types of kernels exist, but no rules exist for selecting an appropriate kernel function. The RBF kernel was selected because it has been shown to exhibit very good performance in a wide variety of applications [22].

5.2.2. Multi-class LS-SVMs

A standard LS-SVM is a binary classifier whereas fault diagnosis is a multi-class problem. Several techniques exist for decomposing a multi-class problem with C classes into a series of binary classification problems (*binarization*). The two main binarization techniques are *One-versus-One* (OvO) and *One-versus-All* (OvA).

The OvO approach involves training a binary classifier for each pair of fault classes. A new input vector $\mathbf{x}_t$ is assigned to the class that is indicated by the majority of the $C(C-1)/2$ individual classifiers. If two or more classes get equal votes, $\mathbf{x}_t$ is unclassifiable.

The OvA method entails training C classifiers to discriminate each fault class from the remaining training data (all other classes combined). The cth classifier labels a point $\mathbf{x}_t$ as either "class C" or "the rest of the data". If $\mathbf{x}_t$ is assigned to two or more classes or to none of the classes, it is deemed unclassfiable. OvA requires fewer binary classifiers to be trained than OvO

Journal homepage: http://www.journals.elsevier.com/journal-of-process-control/
Original file available at: http://www.sciencedirect.com/science/article/pii/S0959152415000177

(C vs. $C(C-1)/2$). As a consequence, the amount of training data per binary classifier is higher for OvA. However, unclassifiable regions are larger and class boundaries tend to be more complex.

In this work, both OvO and OvA are investigated.

5.2.3. Practical implementation

The multi-class LS-SVMs are trained using the `LS-SVMlab` toolbox [25] for MATLAB® available at `http://esat.kuleuven.be/sista/lssvmlab`. `LS-SVMlab` employs a two-step optimization procedure to find the optimal parameters (kernel and regularization parameters) by minimizing the cross-validation error. Coupled Simulated Annealing determines suitable parameters which are subsequently fine tuned with a simplex method.

6. Validation procedure

The proposed method is validated following the steps depicted in Fig. 1. The first step entails the definition of fault classes, i.e., determining the scope of the classification problem. In this case study, the fault classes are given in the first column of Table 2. Note that fault magnitude, sign and occurrence time are of no interest. The scope of the classifier is solely the determination of the fault type.

Step two involves the collection, selection and pretreatment of training data. For this purpose, the generated faulty batches are monitored using the developed NOC PCA model and corresponding fault detection statistics. An alarm signal is raised when a statistic exceeds its control limit at three or more consecutive time points. For the case studies in this paper, all faults are detected less than 10 sample points after their onset. For each faulty batch, the moment of detection is recorded and the data prior to the alarm signal of the fastest statistic (SPE or T^2) are discarded. The remaining data are pretreated using the methods described in Section 2.2.

After the collection and pretreatment steps, the different classifiers are trained. Different sizes of the training set are tested containing 2, 4, 6, 8, or 10 batches per fault class. For each size of the training set, the classifier identification is repeated 100 times, each time using a different set of training batches. For the LS-SVM classifiers, the probabilistic optimization routine employed in their identification yields slightly different values for the regularization parameter and/or kernel parameter for each binary classifier and, hence, slightly different results. Therefore, each LS-SVM identification

is repeated 100 times for a given set of training batches. No such extra repetitions are required for the k-NN classifier as the identification is purely deterministic.

Every trained classifier is subsequently validated on the independent validation set of 100 batches per fault class and the mean values and standard deviations of the correct classification rates are computed. The computed standard deviations mainly reflect the variability due to the 100 different training sets per training set size, as the variability due to the probabilistic LS-SVM parameter optimization routine is significantly lower.

7. Results and discussion

This section validates the proposed method on faulty batches generated from the `Pensim` model. The influence of data pretreatment, binarization method, kernel and size of the training set on the global diagnosis performance (Section 7.1) and class-specific diagnosis performance (Section 7.2) is investigated.

7.1. Global classification rates

Fig. 3 depicts the global correct classification rates (i.e., the mean of the correct classification rate over the six fault classes) for the six different classifiers in the advanced case study with additional measurement noise. The base case study with standard `Pensim` noise shows similar results and is presented in Appendix A.

Five combinations of data transformations and window width are investigated. The first combination ($\triangle$) applies no scaling and uses a data window of one sample which comes down to supplying only the $1 \times J$ vector of the raw measurements at the moment of detection to the classifier. The second combination ($\square$) employs normalization around the mean trajectory followed by auto-scaling of the variable wise unfolded data. It supplies the classifier with the normalized measurement vector at the time of detection. The third combination ($\Diamond$) takes the absolute value of each element of this normalized vector before passing it to the classifier. The fourth combination ($\bigtriangledown$) widens the data window of the previous pretreatment method to five samples. It concatenates the absolute values of the normalized measurement vectors from the moment of detection until four subsequent time points into a $1 \times 5J$ vector. The fifth and last combination ($\bigcirc$) employs a data window of 10 samples.

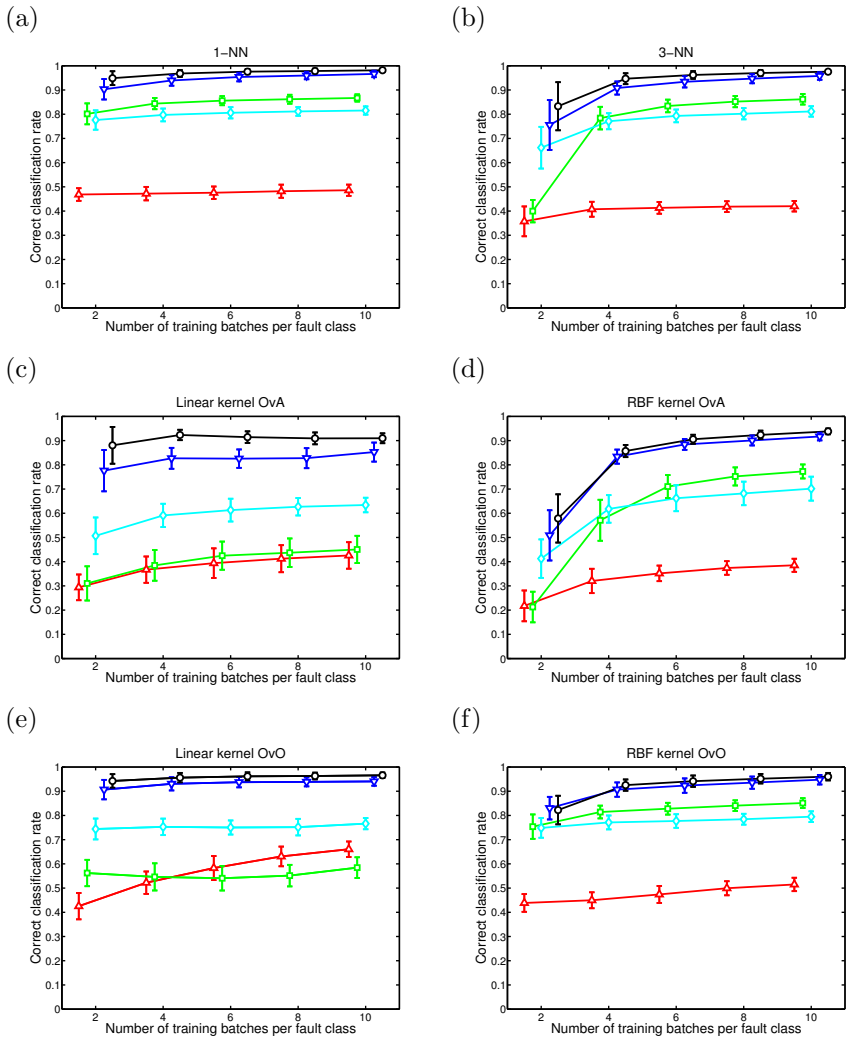

Figure 3: Global correct classification rates in function of the number of training batches for the advanced case study with additional measurement noise using (a) a 1-NN classifier, (b) a 3-NN classifier, (c) an LS-SVM with linear kernel and OvA binarization, (d) an LS-SVM with RBF kernel and OvA binarization, (e) an LS-SVM with linear kernel and OvO binarization, and (f) an LS-SVM with RBF kernel and OvO binarization. The training set consists of raw data ($\triangle$), normalized data ($\square$), the absolute values of the normalized data ($\Diamond$), the absolute values of a window of 5 normalized data points ($\bigtriangledown$) and the absolute values of a window of 10 normalized data points ($\bigcirc$). The total width of each error bar equals two times the standard deviation. The markers are slightly scattered with respect to the x-axis for visual clarity.

Journal homepage: http://www.journals.elsevier.com/journal-of-process-control/
Original file available at: http://www.sciencedirect.com/science/article/pii/S0959152415000177

For all combinations of kernel and binarization method, training on raw data (△) yields the lowest global correct classification rates. As a consequence of the transient nature of batch processes, the raw data pattern depends on the moment of detection. This adds additional uninformative variability to the fault patterns and hence lowers the classification performance.

Normalization around the mean trajectory (□) substantially reduces the influence of the fault detection time. This leads to a more uniform fault pattern presented to the classifier. In most cases, this significantly improves the performance of the k-NN and RBF kernel classifiers. For the linear kernel, this effect is less pronounced—in some the linear kernel even experiences a drop in performance.

To increase the classifier performance even more, it is important to realize that four of the six fault classes contain steps or drifts of both positive and negative sign. Hence, their normalized training data exhibit positive and negative deviations within the same fault class. Taking the absolute value of each normalized measurement eliminates this difference, and a better classification would be expected. As shown in Fig. 3, however, this transformation has a small negative impact on the performance of the k-NN and RBF kernel classifiers, but significantly raises the linear kernel's performance. Section 7.2 will provide an explanation for this counter-intuitive observation.

Extending the data window to 5 or 10 time points further heightens the correct classification rates. The reason for this last performance increase will become clear in Section 7.2 by examining the performance on each fault class instead of the global performance.

While data pretreatment has a large influence on the classifier's performance, the choice of the actual classification model (k-NN or LS-SVM, OvO or OvA, linear or RBF kernel) is of lesser importance. When the data are pretreated appropriately, the performance difference becomes negligible. Overall, the two k-NN models and the LS-SVM classifiers with OvO binarization have a better classification rate than the LS-SVM classifiers with OvA binarization. The results also indicate a slightly higher spread on the classification rates when using an RBF kernel, as an extra parameter (the kernel width) needs to be optimized.

Fig. 3 illustrates the influence of the number of training batches on diagnosis performance. As expected, the correct classification increases with the training set size. This tendency is more pronounced for the RBF kernel and OvA binarization than for the linear kernel and OvO binarization.

7.2. Class-specific classification rates

It is important to also study the performance of each fault class separately instead of only considering the global performance because data pretreatment influences individual performances in different ways. Table 3 lists the mean and standard deviation of the correct classification rate for the k-NN classifiers with a training set size of six faulty batches per class for the case study with additional measurement noise. Tables 4 and 5 list the class-specific rates for the LS-SVM classifiers. Appendix B lists the results of the base case study. These tables exhibit the same general trends as Tables 3–5.

The first column of Tables 3–5 contains the diagnosis performance on raw data. In contrast to the other fault classes, faults on the feed concentration and coolant temperature already exhibit a high correct classification rate. These faults occur at a fixed time point (see Table 2). Hence, their data patterns are unaffected by largely differing detection times which explains their higher performance.

Normalizing around the mean trajectory significantly increases the correct classification rate of agitator power drops and feed rate drifts for the k-NN classifiers and RBF kernel. The performance for the aeration rate drop and DO sensor drift is also improved, but to a lesser extent. In some cases, however, this overall performance gain comes at the cost of a small decrease in correct classification rates for the feed concentration and/or coolant temperature steps. For the linear kernel, only the diagnosis of agitator power drop and aeration rate drop improves. The drift faults (feed rate and DO sensor) exhibit positive and negative deviations within the same fault class and pose problems for the linear kernel. Taking the absolute value of the normalized data improves diagnosis performance on these faults for the linear kernel but reduces the other classifiers' performance for faults on the aeration rate and DO sensor.

Taking the absolute value of the normalization measurements makes the distinction between the aeration rate drop and DO sensor drift more difficult. These faults both influence the DO, which leads to a partial overlap of their characteristic signatures. By taking the absolute value, the difference between positive drifts on the DO sensor and negative drops on the aeration is eliminated, resulting in increased overlap of the two classes and, hence, the lower global classification performance that was already observed in Section 7.1.

This is confirmed by the confusion matrix presented in Table 6. This table lists the assigned labels for the faults on the aeration rate and DO

Postprint version of paper published in J. Process Control 2015, 26:90–101

Table 3: Correct classification rates for each fault class using a 1-NN and 3-NN classifier, and a training set size of six faulty batches per fault class for the advanced case study with additional measurement noise. Please note that the classification rates do not follow a normal distribution.

	Raw		Normalized		Absolute		Window 5		Window 10	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
1-NN										
Coolant temperature step	96.4%	6.0%	94.9%	2.7%	94.0%	4.1%	94.0%	4.0%	95.4%	4.0%
Agitator power drop	23.5%	6.7%	98.4%	4.3%	98.3%	6.2%	98.4%	4.2%	98.4%	4.3%
Feed rate drift	28.0%	8.0%	99.2%	1.3%	99.8%	3.6%	99.9%	0.3%	100.0%	0.2%
Aeration rate drop	25.6%	7.7%	65.8%	13.2%	50.3%	12.6%	91.6%	7.1%	98.0%	2.9%
DO sensor drift	24.2%	6.8%	65.2%	8.0%	50.2%	13.8%	93.0%	4.4%	95.5%	1.5%
3-NN										
Feed concentration step	95.2%	6.8%	83.6%	10.1%	89.3%	9.7%	97.7%	3.4%	99.6%	1.3%
Coolant temperature step	98.6%	4.3%	94.4%	2.9%	92.8%	3.6%	92.8%	3.3%	93.7%	3.1%
Agitator power drop	10.6%	7.5%	94.4%	7.4%	94.3%	7.5%	94.6%	7.5%	94.5%	7.5%
Feed rate drift	17.3%	8.5%	97.7%	6.5%	99.9%	0.3%	100.0%	0.1%	100.0%	0.1%
Aeration rate drop	14.3%	9.6%	70.8%	16.6%	47.6%	14.5%	84.4%	9.5%	94.8%	5.1%
DO sensor drift	12.0%	8.6%	59.5%	9.3%	51.9%	17.9%	90.8%	7.2%	94.6%	1.7%

Journal homepage: http://www.journals.elsevier.com/journal-of-process-control/
Original file available at: http://www.sciencedirect.com/science/article/pii/S0959152415000177

Postprint version of paper published in J. Process Control 2015, 26:90–101

Table 4: Correct classification rates for each fault class using linear and RBF kernels, OvO binarization and a training set size of six faulty batches per fault class for the advanced case study with additional measurement noise. Please note that the classification rates do not follow a normal distribution.

	Raw		Normalized		Absolute		Window 5		Window 10	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
Linear kernel OvO										
Feed concentration step	88.5%	13.6%	37.2%	12.7%	81.8%	11.8%	93.7%	6.1%	98.3%	3.0%
Coolant temperature step	93.8%	8.2%	83.1%	9.6%	89.6%	5.2%	92.0%	4.1%	94.1%	4.7%
Agitator power drop	62.2%	13.1%	75.3%	12.5%	90.3%	8.1%	89.0%	9.4%	91.5%	8.5%
Feed rate drift	41.9%	12.9%	32.2%	11.1%	95.5%	5.6%	99.2%	1.2%	99.7%	0.6%
Aeration rate drop	40.1%	12.0%	75.6%	15.6%	46.5%	13.5%	94.9%	5.0%	98.2%	2.8%
DO sensor drift	23.1%	8.8%	20.9%	10.1%	46.5%	13.4%	93.6%	2.2%	95.1%	0.7%
RBF kernel OvO										
Feed concentration step	86.8%	12.5%	84.3%	11.0%	86.6%	9.9%	92.0%	8.9%	93.0%	8.0%
Coolant temperature step	86.4%	14.0%	88.5%	6.9%	90.0%	8.7%	87.8%	9.0%	91.3%	7.6%
Agitator power drop	32.8%	7.5%	96.8%	4.8%	97.4%	4.0%	96.8%	4.7%	96.8%	4.5%
Feed rate drift	28.2%	9.4%	94.9%	5.8%	94.7%	7.4%	95.3%	5.3%	95.5%	4.1%
Aeration rate drop	25.2%	8.6%	80.8%	12.9%	47.5%	11.6%	90.4%	9.0%	94.3%	6.5%
DO sensor drift	24.9%	7.5%	51.5%	7.0%	50.0%	14.1%	91.6%	4.6%	94.0%	2.4%

Journal homepage: http://www.journals.elsevier.com/journal-of-process-control/
Original file available at: http://www.sciencedirect.com/science/article/pii/S0959152415000177

Postprint version of paper published in J. Process Control 2015, 26:90–101

Table 5: Correct classification rates for each fault class using linear and RBF kernels, OvA binarization and a training set size of 6 faulty batches per fault class for the advanced case study with additional measurement noise. Please note that the classification rates do not follow a normal distribution.

	Raw		Normalized		Absolute		Window 5		Window 10	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
Linear kernel OvA										
Feed concentration step	51.4%	33.4%	28.7%	23.0%	70.2%	14.0%	76.0%	8.3%	91.9%	5.6%
Coolant temperature step	81.2%	11.5%	90.4%	3.9%	85.2%	4.6%	83.3%	3.9%	88.7%	3.2%
Agitator power drop	72.8%	7.7%	79.9%	7.7%	86.2%	7.9%	71.7%	12.7%	81.5%	10.5%
Feed rate drift	7.9%	13.4%	1.4%	4.7%	94.8%	5.4%	93.6%	5.8%	98.4%	1.2%
Aeration rate drop	19.7%	21.6%	49.9%	22.2%	17.6%	19.8%	82.4%	8.7%	94.2%	5.6%
DO sensor drift	3.4%	6.5%	4.6%	7.7%	13.8%	17.8%	88.4%	5.7%	94.1%	2.5%
RBF kernel OvA										
Feed concentration step	82.9%	14.0%	64.5%	10.1%	74.4%	9.4%	78.9%	7.0%	85.6%	6.4%
Coolant temperature step	90.0%	11.3%	87.8%	5.0%	88.4%	5.4%	85.2%	5.1%	87.7%	4.5%
Agitator power drop	26.0%	7.2%	95.4%	6.4%	97.2%	4.3%	94.5%	5.0%	91.5%	5.4%
Feed rate drift	6.3%	7.3%	89.0%	7.0%	94.7%	5.1%	95.2%	3.0%	94.5%	2.7%
Aeration rate drop	3.2%	5.1%	46.1%	21.4%	20.5%	18.5%	87.4%	8.4%	91.9%	5.2%
DO sensor drift	3.0%	5.3%	43.3%	10.8%	22.0%	20.6%	89.3%	5.5%	92.0%	2.6%

Journal homepage: http://www.journals.elsevier.com/journal-of-process-control/
Original file available at: http://www.sciencedirect.com/science/article/pii/S0959152415000177

Postprint version of paper published in J. Process Control 2015, 26:90–101
The content is identical to the published paper, but without the final typesetting by the publisher.

Table 6: Confusion matrix for the 3-NN approach after normalizing the measurements around their average trajectory and taking the absolute value of the training data. The training set consists of 6 batches per class.

	Actual fault	
Predicted fault	Aeration rate	DO sensor
Aeration rate	47.6%	43.2%
DO sensor	46.9%	51.9%
Other	3.2%	3.7%
Unclassifiable	2.3%	1.3%

sensor when using the absolute value of the normalized data as input for the classifiers. Of the aeration rate faults, 46.9% are wrongly classified as DO sensor faults. Likewise, if the true fault is a DO sensor fault, the classifier predicts a problem with the aeration rate in 43.2% of the cases. Hence, the classifier has difficulties discriminating DO sensor faults from aeration faults.

By observing the time evolution of the DO measurements, it is possible to discriminate between both faults as illustrated in Fig. 4 and improve the performance. Extending the data window to 5 or 10 time points enables the classifier to learn the different time evolution. As a consequence, the correct classification rates in Table 3 exhibit a significant increase for aeration and DO sensor faults.

These results show the importance of employing the correct data pretreatment for the faults that must be diagnosed as some pretreatments improve the classification of some types of process upsets while reducing the correct classification for other faults. An in-depth analysis of a classifier via confusion matrix can help select and tune the data pretreatment to improve the performance for specific types of process faults.

8. Conclusions

Existing approaches for classification-based batch diagnosis are essentially offline diagnosis methods where the classification model is trained on the full data of each faulty batch. However, to obtain a characteristic pattern for each fault class and, hence, improve classification performance, it is important to focus on data obtained at the start of the period of faulty operation. Data

Journal homepage: http://www.journals.elsevier.com/journal-of-process-control/
Original file available at: http://www.sciencedirect.com/science/article/pii/S0959152415000177

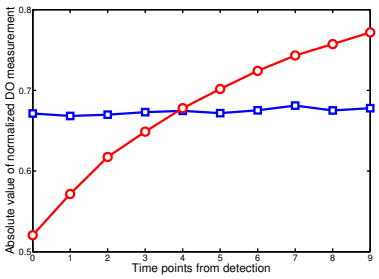

Figure 4: Mean trajectory of the absolute value of the normalized DO measurement during a step decrease in aeration rate (□) and DO sensor drift (◯).

before the onset of the fault are simply NOC data and samples obtained too long after the onset might be influenced by different corrective actions and additional disturbances. Therefore, a novel data selection methodology was proposed which focuses on the onset of the fault, avoiding the estimation of future batch evolution.

Additionally, several data pretreatment steps were proposed (normalizing around the mean trajectory, taking the absolute value and defining a data window) to transform the data into a form suitable for easy classification. Two benchmark studies conducted on the `Pensim` simulator (a base study with the standard `Pensim` noise and one with additional measurement noise) using both simple k-NN and advanced LS-SVM classifiers illustrated the strong impact of the proposed data preprocessing steps on diagnosis performance.

The case studies clearly indicated that while overall classification performance might increase by changing the data pretreatment, this might also result in a lower correct classification rate for some types of faults. Analysis of the class-specific correct classification rates and the classifier's confusion matrix can help select the correct pretreatment for a given set of process faults. These tools can also be used to optimize the correct classification rates of the most important (or safety-critical) faults.

While data selection and pretreatment had a large influence on the classifier performance, the choice between the actual classification model (k-NN or LS-SVM, OvO or OvA binarization technique, linear or RBF kernel) was often of lesser importance. Varying the number of training batches demonstrated that satisfactory diagnosis performance is achievable even for low

amounts of training data, provided the data are preprocessed appropriately.

Acknowledgements

Work supported in part by Project PFV/10/002 (OPTEC Optimization in Engineering Center) of the Research Council of the KU Leuven, the KU Leuven knowledge platform SCORES4CHEM (www.scores4chem.be), and the Belgian Program on Interuniversity Poles of Attraction initiated by the Belgian Federal Science Policy Office. P. Van den Kerkhof and J. Vanlaer are funded by a Ph.D. grant of the agency for Innovation by Science and Technology (IWT). The authors assume scientific responsibility.

Appendix A. Additional global classification results

Fig. A.5 depicts the global correct classification rates (i.e., the mean of the correct classification rate over the six fault classes) for the six different classifiers in the base case study with standard `Pensim` measurement noise.

Appendix B. Additional class-specific results

Tables B.7–B.9 present the class-specific classification rates for the base case study with standard `Pensim` noise. Tables 4 and 5 present the class-specific classification rate for the various LS-SVM classifiers for the advanced case study with additional measurement noise.

References

1. Chen, J., Liu, K.. On-line batch process monitoring using dynamic PCA and dynamic PLS models. *Chem Eng Sci* 2002;**57**(1):63–75.

2. Choi, S., Morris, J., Lee, I.B.. Dynamic model-based batch process monitoring. *Chem Eng Sci* 2008;**63**:622–636.

3. Lee, J., Yoo, C., Lee, I.. Statistical monitoring of dynamic processed based on dynamic independent component analysis. *Chem Eng Sci* 2004; **59**:2995–3006.

4. Lee, J., Yoo, C., Choi, S., Vanrolleghem, P., Lee, I.. Nonlinear process monitoring using kernel principal component analysis. *Chem Eng Sci* 2004;**59**:223–234.

5. Wang, L., Shi, H.. Multivariate statistical process monitoring using an improved independent component analysis. *Chem Eng Res Des* 2010; **88**:403–414.

6. Qin, S.. Survey on data-driven industrial process monitoring and diagnosis. *Ann Rev Control* 2012;**36**:220–234.

7. Miller, P., Swanson, R., Heckler, C.. Contribution plots: a missing link in multivariate quality control. *Appl Math Comp Sci* 1998;**8**(4):775–792.

8. MacGregor, J., Jaeckle, C., Kiparissides, C., Koutoudi, M.. Process monitoring and diagnosis by multiblock PLS methods. *AIChE J* 1994; **40**(5):826–838.

9. Westerhuis, J., Gurden, S., Smilde, A.. Generalized contribution plots in multivariate statistical process monitoring. *Chemom Intell Lab* 2000; **51**(1):95–114.

10. Van den Kerkhof, P., Vanlaer, J., Gins, G., Van Impe, J.. Analysis of smearing-out in contribution plot based fault isolation for statistical process control. *Chem Eng Sci* 2013;**104**:285–293.

11. Chiang, L., Russell, E., Braatz, R.. Fault diagnosis in chemical processes using Fisher discriminant analysis, discriminant partial least squares, and principal components analysis. *Chemom Intell Lab* 2000; **50**:243–252.

12. Chiang, L., Kotanchek, M., Kordon, A.. Fault diagnosis based on Fisher discriminant analysis and support vector machines. *Comput Chem Eng* 2004;**28**:1389–1401.

13. Yélamos, I., Escudero, G., Graells, M., Puigjaner, L.. Performance assessment of a novel fault diagnosis system based on support vector machines. *Comput Chem Eng* 2009;**33**:244–255.

14. Mahadevan, S., Shah, S.. Fault detection and diagnosis in process data using one-class support vector machines. *J Process Control* 2009; **19**:1627–1639.

15. Ruiz, D., Nougués, J., Calderón, Z., na, A.E., Puigjaner, L.. Neural network based framework for fault diagnosis in batch chemical plants. *Comput Chem Eng* 2000;**24**(2-7):777–784.

16. Cho, H.W., Kim, K.J.. Fault diagnosis of batch processes using discriminant model. *Int J Prod Res* 2004;**42**(3):597–612.

17. Cho, H.W., Kim, K.J.. Diagnosing batch processes with insufficient fault data: generation of pseudo batches. *Int J Prod Res* 2005; **43**(14):2997–3009.

18. Cho, H.W.. Nonlinear feature extraction and classification of multivariate process data in kernel feature space. *Expert Syst Appl* 2007; **32**:534–542.

19. Li, J., Cui, P.. Improved kernel fisher discriminant analysis for fault diagnosis. *Expert Syst Appl* 2009;**36**:1423–1432.

20. Monroy, I., Villez, K., Graells, M., Venkatasubramanian, V.. Fault diagnosis of a benchmark fermentation process: a comparative study of feature extraction and classification techniques. *Bioprocess Biosyst Eng* 2012;**35**(5):689–704.

21. Vapnik, V.. *Statistical Learning Theory*. New York: John Wiley & Sons; 1998.

22. Abe, S.. *Support Vector Machines for Pattern Classification*. Springer-Verlag; 2005.

23. Birol, G., Ündey, C., Çinar, A.. A modular simulation package for fed-batch fermentation: penicillin production. *Comput Chem Eng* 2002; **26**:1553–1565.

24. Fix, E., Hodges, J.. Discriminatory analysis, nonparametric discrimination: Consistency properties. Tech. Rep. 4; USAF School of Aviation Medicine; Randolph Field, Texas, USA; 1951.

25. Suykens, J., Van Gestel, T., De Brabanter, J., De Moor, B., Vandewalle, J.. *Least Squares Support Vector Machines*. Singapore: World Scientific; 2002.

26. Nomikos, P., MacGregor, J.. Monitoring batch processes using multiway principal component analysis. *AIChE J* 1994;**40**(8):1361–1375.

27. Nomikos, P., MacGregor, J.F.. Multivariate SPC charts for monitoring batch processes. *Technometrics* 1995;**37**(1):41–59.

682 28. Bissessur, Y., Martin, E., Morris, A.. Monitoring the performance of
683 the paper making process. *Control Eng Pract* 1999;**7**:1357–1368.

684 29. Dong, D., McAvoy, T.. Nonlinear principal component analysis-based
685 on principal curves and neural networks. *Comput Chem Eng* 1996;**20**:65–
686 78.

687 30. Choi, S., Park, J., Lee, I.. Process monitoring using a Gaussian mix-
688 ture model via principal component analysis and discriminant analysis.
689 *Comput Chem Eng* 2004;**28**:1377–1387.

690 31. Simoglou, A., Georgieva, P., Martin, E., Morris, A., Feyo de Azevedo,
691 S.. On-line monitoring of a sugar crystallization process. *Comput Chem*
692 *Eng* 2005;**29**:1411–1422.

693 32. Lennox, B., Montague, G., Hiden, H., Kornfeld, G., Goulding, P..
694 Online estimation of biomass, glucose and ethanol in *Saccharomyces*
695 *cerevisiae* cultivations using in-situ multi-wavelength fluorescence and
696 software sensors. *J Biotechnol* 2009;**144**:125–135.

697 33. Gins, G., Vanlaer, J., Van Impe, J.. Discriminating between critical
698 and non-critical disturbances in (bio-)chemical batch processes using
699 multi-model fault detection and end-quality prediction. *Ind Eng Chem*
700 *Res* 2012;**51**(38):12375–12385.

701 34. Van den Kerkhof, P., Gins, G., Vanlaer, J., Van Impe, J.. Dynamic
702 model-based fault diagnosis for (bio)chemical batch processes. *Comput*
703 *Chem Eng* 2012;**40**:12–21.

704 35. Ge, Z., Song, Z., Gao, F.. Review of recent research on data-based
705 process monitoring. *Ind Eng Chem Res* 2013;**52**:3543–3562.

706 36. Jollife, I.. *Principal Component Analysis*. Springer Verlag; 1986.

707 37. Jackson, J.. *A User's Guide to Principal Component Analysis*. John
708 Wiley & Sons; 1991.

709 38. Nomikos, P., MacGregor, J.F.. Multi-way partial least squares in
710 monitoring batch processes. *Chemom Intell Lab* 1995;**30**:97–108.

711 39. Wold, S., Kettaneh, N., Fridén, H., Holmberg, A.. Modelling and di-
712 agnosis of batch processes and analogous kinetic experiments. *Chemom*
713 *Intell Lab* 1998;**44**:331–340.

714 40. Li, B., Morris, J., Martin, E.. Model selection for partial least squares
715 regression. *Chemom Intell Lab* 2002;**64**(1):79–89.

716 41. Stone, C.. Consistent nonparametric regression. *Ann Stat* 1977;**5**:595–
717 620.

718 42. Devroye, L., Gyorfy, L., Lugosi, G.. *A Probabilistic Theory of Pattern*
719 *Recognition*. Springer; 1996.

720 43. Wu, X., Kumar, V., Quinlan, J., Ghosh, J., Yang, Q., Motoda,
721 H., et al. Top 10 algorithms in data mining. *Knowl Inf Syst* 2008;
722 **14**(1):1–37.

723 44. King, R., Feng, C., Sutherland, A.. StatLog: Comparison of classifi-
724 cation algorithms on large real-world problems. *Appl Artif Intell* 1995;
725 **9(3)**:289–333.

726 45. Beyer, K., Goldstein, J., Ramakrishnan, R., Shaft, U.. When is
727 "nearest neighbor" meaningful? In: Beeri, C., Buneman, P., editors.
728 *Proceedings of the 7th International Conference on Database Theory*.
729 Jerusalem (Israel); 1999, p. 217–235.

730 46. Bezergianni, S., Kalogianni, A.. Application of principal component
731 analysis for monitoring and disturbance detection of a hydrotreating
732 process. *Ind Eng Chem Res* 2008;**47**:6972–6982.

Postprint version of paper published in J. Process Control 2015, 26:90–101
The content is identical to the published paper, but without the final typesetting by the publisher.

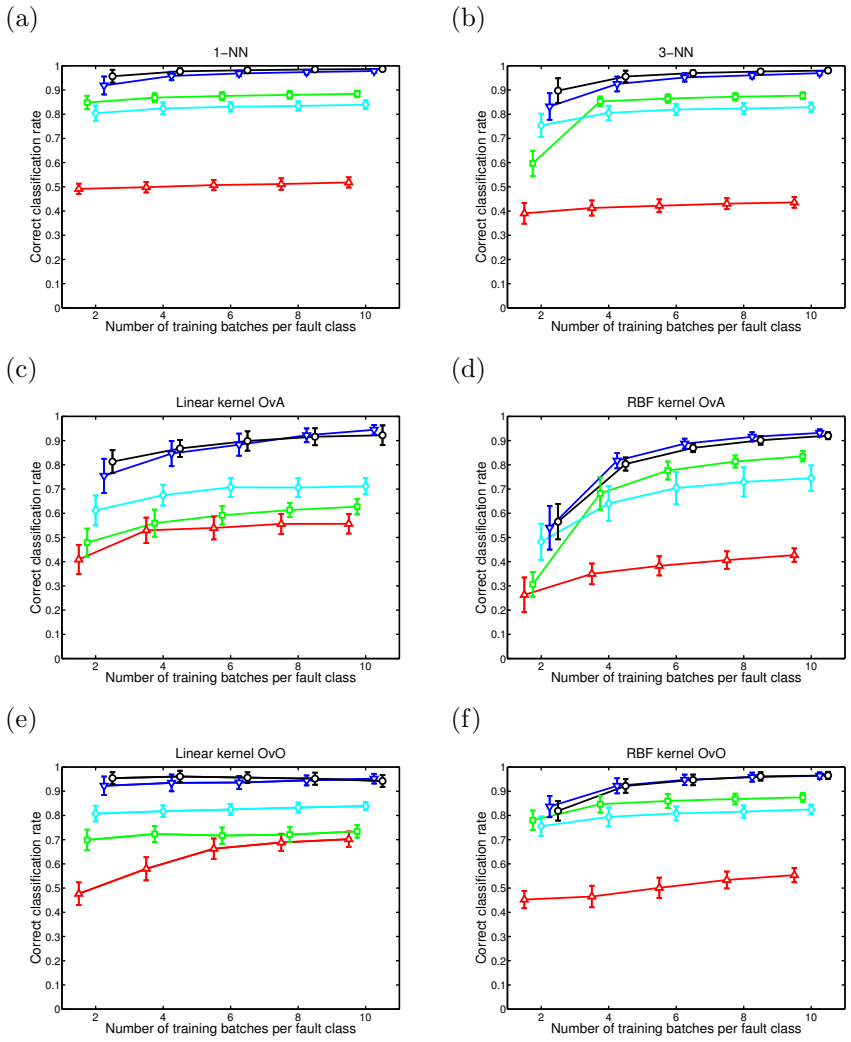

Figure A.5: Global correct classification rates in function of the number of training batches for the base case study with the standard `Pensim` noise using (a) a 1-NN classifier, (b) a 3-NN classifier, (c) an LS-SVM with linear kernel and OvA binarization, (d) an LS-SVM with RBF kernel and OvA binarization, (e) an LS-SVM with linear kernel and OvO binarization, and (f) an LS-SVM with RBF kernel and OvO binarization. The training set consists of raw data (△), normalized data (□), the absolute values of the normalized data (◇), the absolute values of a window of 5 normalized data points (▽) and the absolute values of a window of 10 normalized data points (○). The total width of each error bar equals two times the standard deviation. The markers are slightly scattered with respect to the x-axis for visual clarity.

Journal homepage: http://www.journals.elsevier.com/journal-of-process-control/
Original file available at: http://www.sciencedirect.com/science/article/pii/S0959152415000177

Table B.7: Correct classification rates for each fault class using a 1-NN and 3-NN classifier, and a training set size of six faulty batches per fault class for the base case study with standard `Pensim` noise. Please note that the classification rates do not follow a normal distribution.

	Raw		Normalized		Absolute		Window 5		Window 10	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
1-NN										
Feed concentration step	95.1%	8.0%	100.0%	0.0%	100.0%	0.0%	100.0%	0.0%	100.0%	0.0%
Coolant temperature step	100.0%	0.0%	90.6%	5.0%	91.0%	5.1%	97.2%	1.9%	97.7%	1.6%
Agitator power drop	28.9%	6.7%	98.1%	4.5%	98.3%	4.5%	98.1%	4.5%	98.1%	4.5%
Feed rate drift	30.3%	7.8%	99.9%	0.4%	100.0%	0.0%	100.0%	0.0%	100.0%	0.0%
Aeration rate drop	25.5%	7.5%	64.8%	12.2%	50.1%	10.0%	94.5%	5.0%	98.4%	4.3%
DO sensor drift	24.6%	5.6%	71.8%	8.6%	59.4%	12.1%	91.2%	3.7%	94.9%	1.7%
3-NN										
Feed concentration step	99.6%	1.6%	100.0%	0.1%	100.0%	0.0%	100.0%	0.0%	100.0%	0.0%
Coolant temperature step	100.0%	0.0%	85.6%	4.4%	85.7%	4.4%	93.5%	3.5%	95.0%	2.5%
Agitator power drop	13.3%	7.9%	93.8%	6.2%	93.8%	6.2%	93.9%	6.2%	94.0%	6.1%
Feed rate drift	17.1%	8.6%	99.8%	0.6%	100.0%	0.0%	100.0%	0.1%	100.0%	0.1%
Aeration rate drop	11.9%	8.5%	72.0%	15.8%	48.2%	14.4%	93.8%	5.6%	99.2%	1.7%
DO sensor drift	11.3%	6.1%	67.7%	10.4%	64.1%	17.5%	89.9%	4.5%	94.0%	1.2%

Postprint version of paper published in J. Process Control 2015, 26:90–101

Table B.8: Correct classification rates for each fault class using linear and RBF kernels, OvO binarization and a training set size of six faulty batches per fault class for the base case study with standard `Pensim` noise. Please note that the classification rates do not follow a normal distribution.

	Raw		Normalized		Absolute		Window 5		Window 10	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
Linear kernel OvO										
Feed concentration step	97.9%	5.3%	100.0%	0.1%	100.0%	0.3%	99.3%	1.2%	99.9%	0.6%
Coolant temperature step	99.7%	1.9%	94.9%	4.7%	98.1%	4.1%	81.3%	14.3%	87.8%	11.6%
Agitator power drop	73.1%	10.2%	82.0%	7.0%	94.7%	6.2%	95.5%	6.0%	96.2%	5.9%
Feed rate drift	50.3%	13.3%	47.1%	11.2%	96.1%	3.6%	97.0%	3.6%	97.1%	3.3%
Aeration rate drop	49.2%	14.2%	75.1%	15.8%	52.2%	12.2%	95.9%	4.1%	98.1%	3.0%
DO sensor drift	27.4%	9.0%	31.0%	9.5%	53.7%	13.8%	92.6%	3.5%	94.8%	1.7%
RBF kernel OvO										
Feed concentration step	87.8%	14.1%	94.3%	8.5%	93.8%	8.6%	94.9%	6.4%	95.1%	6.3%
Coolant temperature step	90.8%	14.2%	91.7%	5.2%	92.5%	5.2%	98.5%	1.9%	97.2%	4.2%
Agitator power drop	41.5%	10.1%	93.0%	10.4%	93.2%	10.6%	91.2%	9.4%	89.7%	9.9%
Feed rate drift	29.5%	8.1%	97.9%	2.9%	96.8%	5.7%	98.2%	3.4%	97.8%	4.4%
Aeration rate drop	26.0%	7.7%	77.7%	15.9%	50.1%	11.4%	95.1%	5.3%	96.4%	4.8%
DO sensor drift	25.0%	7.6%	61.4%	9.0%	58.7%	14.5%	89.8%	5.0%	92.0%	4.0%

Journal homepage: http://www.journals.elsevier.com/journal-of-process-control/
Original file available at: http://www.sciencedirect.com/science/article/pii/S0959152415000177

Postprint version of paper published in J. Process Control 2015, 26:90–101

Table B.9: Correct classification rates for each fault class using linear and RBF kernels, OvA binarization and a training set size of six faulty batches per fault class for the base case study with standard `Pensim` noise. Please note that the classification rates do not follow a normal distribution.

	Raw		Normalized		Absolute		Window 5		Window 10	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
Linear kernel OvA										
Feed concentration step	96.5%	6.4%	98.2%	2.7%	96.8%	1.9%	94.2%	5.2%	98.8%	1.6%
Coolant temperature step	99.6%	1.7%	93.8%	5.5%	95.6%	5.6%	64.4%	21.1%	56.7%	20.6%
Agitator power drop	78.3%	5.8%	84.6%	4.7%	91.9%	6.2%	90.0%	7.8%	92.3%	6.2%
Feed rate drift	5.2%	8.8%	2.6%	7.5%	97.1%	2.9%	97.0%	3.9%	99.2%	1.5%
Aeration rate drop	39.6%	25.8%	65.8%	21.2%	33.2%	21.8%	92.6%	6.1%	98.0%	3.0%
DO sensor drift	4.4%	7.8%	10.0%	12.5%	9.7%	16.5%	91.7%	3.4%	94.0%	1.9%
RBF kernel OvA										
Feed concentration step	89.6%	12.3%	95.8%	4.1%	95.3%	5.0%	96.2%	3.1%	96.2%	2.5%
Coolant temperature step	94.4%	11.2%	78.4%	8.3%	79.2%	8.6%	72.0%	5.2%	62.5%	6.1%
Agitator power drop	32.4%	9.0%	88.6%	9.8%	88.8%	10.3%	84.4%	9.9%	80.8%	9.8%
Feed rate drift	5.2%	6.5%	93.5%	4.9%	96.5%	5.1%	97.6%	2.9%	96.1%	3.4%
Aeration rate drop	4.5%	6.3%	57.2%	17.5%	27.7%	18.6%	93.7%	5.1%	95.3%	3.5%
DO sensor drift	3.6%	4.6%	52.4%	8.0%	34.9%	23.3%	88.5%	5.6%	91.1%	3.0%

Journal homepage: http://www.journals.elsevier.com/journal-of-process-control/
Original file available at: http://www.sciencedirect.com/science/article/pii/S0959152415000177